The Challenges of Extract, Transform and Loading (ETL) System Implementation For Near Real-Time Environment

A Back Room Staging for Data Analytics

Adilah Sabtu*[1,2], Nurulhuda Firdaus Mohd Azmi[1,2], Nilam Nur Amir Sjarif[1,2], Saiful Adli Ismail[1,2], Othman Mohd Yusop[1], Haslina Sarkan[1], Suriayati Chuprat[1]

[1]Advanced Informatics School (UTM AIS)
[2]Machine Learning for Data Science Interest Group (MLDS)
Universiti Teknologi Malaysia (UTM)
Jalan Sultan Hj Yahya Petra, 54100 Kuala Lumpur, Malaysia.
{adilah.sabtu@gmail.com, |huda, nilamnur, saifuladli, haslinams, suriayati.kl|@utm.my}

***Abstract*— Organization with considerable investment into data warehousing, the influx of various data types and forms requires certain ways of prepping data and staging platform that support fast, efficient and volatile data to reach its targeted audiences or users of different business needs. Extract, Transform and Load (ETL) system proved to be a choice standard for managing and sustaining the movement and transactional process of the valued big data assets. However, traditional ETL system can no longer accommodate and effectively handle streaming or near real-time data and stimulating environment which demands high availability, low latency and horizontal scalability features for functionality. This paper identifies the challenges of implementing ETL system for streaming or near real-time data which needs to evolve and streamline itself with the different requirements. Current efforts and solution approaches to address the challenges are presented. The classification of ETL system challenges are prepared based on near real-time environment features and ETL stages to encourage different perspectives for future research.**

Keywords— ETL; streaming data; near real-time environment; high availability; low latency; horizontal scalability

I. INTRODUCTION

In the spectrum of big data, per Forrester Research Q1 2016 report, data preparation and data integration technologies are within the gamut of intense survival and growth phases respectively. Ultimately, the goal is to make data flow fast, fresh and seamless across multiple sources in a data warehouse and provide sound bases to accommodate business intelligence or analytics.

Less glamorous than data analytics yet an enabler with equally important stake, data integration involves vetting data across many databases with different business demands to produce a centralized master data management system among other operational systems. Normally, data are integrated before operational and transactional activities. The effectiveness of data integration is basically vindicated by how well the data deliver and realize it's intended of use, which in most cases are for analytics. Data warehouse commonly engages extract, transform and load (ETL) system for maneuvering the operational structuring and transaction of the data. Traditional or conventional ETL is comprised of extract, transform and load stages where the steps involved loosely follow the order of the abbreviated term.

A typical ETL system may prove to be effective in managing structured, batch-oriented data which, up to date and within scope for corporate insights and decision making. However, dealing with faster stream, time sensitive or sensor data requires different model and rather massive tweaking to the ETL system [1] where high availability, low latency and horizontal scalability are 3 key features that need to be addressed in near real-time environment [2]. In this environment, data need to be available fresh, pronto and continuously.

This paper aims to expand existing research and classify the challenges and measures in developing ETL system for near real-time or streaming data to provide guidance and focus for research and backroom developers of data analytics. The structure of the paper is as follows: Section II provides an overview of the ETL system. Later, Section III discusses the three (3) key features of near real-time environment, which are high availability, low latency and horizontal scalability. Next, the challenges for near real-time or streaming data encountered within an ETL system based on these features is discussed in Section IV. Section V discussed about works on ETL systems implemented for near real-time data warehousing.

II. ETL SYSTEM OVERVIEW

ETL System is a broad presentation of data movement and transactional processes from extraction of multiple application sources, transforming data into dimensional fact table and conformed format to feed data into data warehouse environment such as data marts, data stores or other target

978-1-5090-6255-3/17/$31.00 ©2017 IEEE

systems [1] [3]. The process is widely applied in data integration, migration, staging and master data management efforts. Fig. 1 illustrates a basic ETL system flow in a data warehouse environment. The flow involved accessing data from source locations, then, ETL processing which included cleaning, integrating and staging the data before transporting and channeling the transformed data to target systems.

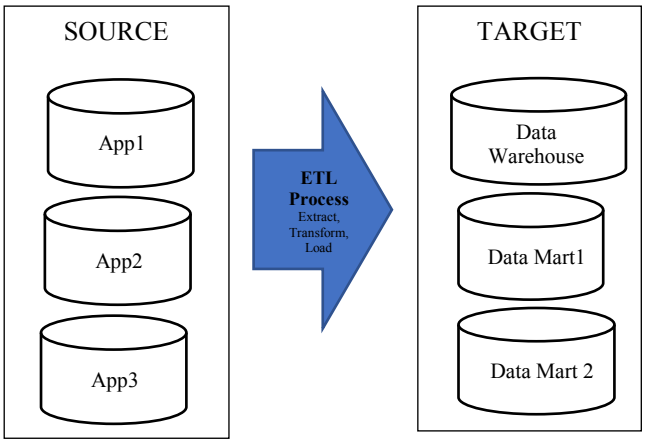

Fig. 1. ETL System Flow in a Data Warehouse Environment

Like any living sentinels which in time experience change and evolvement, a living IT system also needs to undergo evolution to ensure its relevance and continuity in the face of environment change. The management and maintenance part of ETL system need to have the capacity to address change of directions and to support new data demand which are becoming more huge volume, endless streaming, wider variety of types and sources, request for real-time user requirements and new analytics, extreme integration, availability of new technologies and more powerful tools [4].

III. Key Features of Near Real-Time Environment

Near real-time environment is mainly featured by its high availability of streaming data, low latency of intervals between transactions and horizontal scalability for performance improvement. Failing to address these features would affect or limit functionality of an operational system:

A. High Availability

Streaming data as the name dictates are always available in endless constant flow within mere seconds. They are sensitive by nature that the slightest disruption would affect operations.

Distribution and replication are important considerations to ensure fluidity of data collection process, loss data can be recovered and always available when called upon, even when pitted against outage, data loss or receiving overloads of throughput [2].

Unlike batch data which are distributed periodically, change of fresh data is less frequent and the same batch data can be used repeatedly.

B. Low Latency

The requirement for speed in delivering fresh data to meet business needs is called data latency. Compared to periodical requirement of batch data, the time between the events where data arrive and data made available to user for near real-time data is almost instant and low latency [2].

Traditional ETL was designed to accommodate batch processing where data refreshment can be done during off peak hours of which operational activities can be temporarily suspended [5] therefore could not produce accurate result for near real-time analysis where stream data flow continuously.

C. Horizontal Scalability

Horizontal scaling is a sustainable approach for improving performance and mitigating the risk of interruption of seamless data flow [2]. It adds separate independent servers to a cluster of servers handling different tasks and requires only low coordination between systems.

This type of scaling offers higher potential of expansion and less risky measures compared to vertical scaling where boosting its performance is limited within the capacity of a single machine. If the machine flopped, data flow is interrupted and the system could lose pertinent data.

IV. Challenges Within The Stages of ETL For Near Real-Time Environment

As the environment for near real-time data lean more on the extreme side; incessant flow of fresh data in voluminous amount and real-time reporting, traditional ETL system requires major works and remodeling to support the requirement. The identified ETL challenges and solution approaches based on environment features are presented in Table 1, Table 2 and Table 3. The details of these solutions with the existing studies related to the identified key features of near real-time environment will be discussed in Section IV.

A. Extraction

The process of extraction involves profiling data dimensional attributes from separate databases, capturing change data and getting data from the source location. Streaming data such as network traffics, click streams and sensors are fleeting, constantly changing and continuous. It challenges the way change data are captured for constant update and loaded without becoming overloaded and disrupting normal operational activities [6] [5]. Table 1 provides the overview of extraction challenges and the solution approaches when it is implemented for near real-time environment.

TABLE 1: Overview of Challenges in Extract Stage For Near Real-Time Environment

Feature	Extract Stage	
	Challenge	***Solution Approach***
High Availability	a) Heterogenous Data Source b) Backup Data	a) Stream Processor, Semantic web technologies toolkits

Feature	Extract Stage	
	Challenge	***Solution Approach***
	c) OLAP Internal Inconsistency	b) Server for replication; Log-based CDC c) Snapshot; RTDC; Layer-based View; RODB
Low Latency	a) Multiple Data Source Integration b) Data Source Overload	a) Combine change data capture, stream processor and data integration tools b) Update significance and record changed method; Special format for CDC log
Horizontal Scalability	-	-

B. Transformation

Transformation is the process where data are cleaned and conformed into fact table or predetermined format which can be shared across different business platforms and needs. The challenges posed in this process is that master data is not compromised when joined with transactional data [5]. For near real-time data, transactional data are constantly refreshed so the frequency of having to update master data too is higher than batch data.

In addition, the amount of data carried into the warehouse after transformation is smaller and also constant that made the transformation process quite inefficient by processing smaller amount at more frequent rate [5]. Table 2 portrays the overview of challenges in transform stage for near real-time environment.

TABLE 2: OVERVIEW OF CHALLENGES IN TRANSFORM STAGE FOR NEAR REAL-TIME ENVIRONMENT

Feature	Transform Stage	
	Challenges	***Solution Approach***
High Availability	-	-
Low Latency	a) Master Data Overhead b) Intermediate Server For Aggregation	a) Master data cache + data base queue b) ELT (Extract Load Transform)
Horizontal Scalability	Separate Server For Aggregation	ELT (Extract Load Transform)

.

C. Loading

In loading stage, the transformed data or metadata are delivered into the warehouse target dimensional models or tables. The challenges are to maintain optimum performance during OLAP or online analytical processing to avoid overlap while loading [5] [7] and due to OLAP internal inconsistency [8] [9]. Summary on the challenges in loading stage for near real-time environment is in Table 3.

TABLE 3: OVERVIEW OF CHALLENGES IN LOADING STAGE FOR NEAR REAL-TIME ENVIRONMENT

Feature	Loading Stage	
	Challenges	***Solution Approach***
High Availability	OLAP Internal Inconsistency	Snapshot; RTDC; Layer-based View; RODB; Dynamic mirror
Low Latency	Performance Degradation	Staging Table; Multi Stage Trickle & Flip
Horizontal Scalability	OLAP Internal Inconsistency	Staging OLAP outside data warehouse update period; CR-OLAP

IV. RELATED WORKS ON NEAR REAL-TIME ENVIRONMENT

Existing research classified the challenges and solutions in different ways such as ETL stages [5], real-time modeling, real-time mechanism, real-time ETL enabler, OLAP issues [8] and data stream management system [10]. This paper explored the related works based on three (3) key environment features such as high availability, low latency and horizontal scalability. These features have been discussed in previous section.

A. High Availability

Handling heterogeneous data sources might require separate kinds of processors and configurations to ensure that fresh data and the necessary backup were always at hand, such as streaming processor [11] for streaming data and change data capture (CDC) for stored data [5]. Setting up separate server for backup and replication or CDC log-based technique [6] ensures availability that the risk of data loss is mitigated [2].

Other solution applied semantic web technologies toolkits or programs that support real-time streaming in ETL system to leverage connections between disparate data sources [3] and [12]. Tested in an automotive environment, the inference methods, iterative process and expanded ontology federated process used over the technologies helped with the identification of relationship among the heterogeneous data sources and were expected to support data integration initiatives [13].

OLAP allowed different kinds of multidimensional analytics. Having no mechanism from preventing analysis and update happening concurrently, the result might not be accurate. A dynamically generated layer-view combined with lock row where new layer is automatically generated for every new transactional occurrence or new analysis helped to ensure that the latest data is provided [8]. Other plausible solutions are staging data for data analysis separate from the update session of the data warehouse, taking snapshots of the data warehouse table or using Real-Time Data Cache (RTDC) [10] [5]. Fang et al. [14] used Real-Time Operational Database (RODB) in the middleware layer to effectively manage massive sensor data and devices and storage. Dynamic mirror technology solution was proposed by Li and Mao [15] to manage OLAP queries and Online Transaction Processing (OLTP) updates from blocking each other functionality.

B. Low Latency

Multiple data source integration could also consider separate kinds of processors and configurations to ensure streaming fresh data such as streaming processor and change data capture (CDC) integrated using data integration tools [6] [5]. CDC techniques can adequately contain data source overload using special format log although there are still latency between the original and captured data [16]. Problem when using CDC Trigger technique such as master data overhead was overcome by maintaining separate queues, placing master data in a cache and real-time data in a database queue [6] [5].

Narendra et.al [17] proposed a form of ELT approach called data warehouse-based data model for analytics which need a fraction of streaming data only at a time. This approach filtered data required for analysis and extracted them into a staging database of server for aggregation. Then, the data were loaded into the warehouse for transformation. This solution overcomes the problem of reduced speed if a large amount of data is processed and stored.

Jain et.al [6] compared many CDC techniques which support real-time data. Log-based technique was considered a better technique based on the minimal impact on source database. Its log files retention minimized data loss, reduced performance degradation and sped up analysis process. In addition, two (2) critical data identifying measures, update significance and record changed methods were used to prevent data source overload. Further study by Jain et.al [6] suggested staging temporary tables following an exact replica of the data warehouse destination tables for receiving and storing new data. Further study by Valencio et.al [18] used a technique termed as Real Time Delta Extraction based on Triggers and Multi-stage Trickle and flip technique by Zuters [7] basically adapted and combined conventional and real-time ETL techniques aimed for zero latency when processing constant small data loads.

C. Horizontal Scalability

Narendra et.al [17] showed the effectiveness of scalability in a system architecture built on different modules, servers and process processors to perform each own separate goals, processes, functions and scheduling when feeding the many requirements of near real-time data. Another solution successfully staged OLAP outside the data warehouse update period [14] [5].

Another research [19] introduced CR-OLAP a cloud-based real-time OLAP system utilized cloud infrastructure consisting multi-core processors which increased database size and maintain performance

V. CONCLUSION

Data integration of Extract, transform, and load (ETL) technologies has been used to accomplish this in traditional data warehouse environments. However, as we enter the Brontobyte Age, traditional approaches of data integration such as ETL begin to show their limits. Thus, traditional ETL needs to be enhanced to support streaming of data processes with federation across multiple sources. The role of ETL needs to be evolving to handle newer data management environments. Traditional or conventional ETL System is facing challenges in keeping up with the evolving requirements of big data in data warehousing. This paper reviewed existing literatures and classified the challenges in ETL system for streaming or near real-time data based the environment key features which are high availability, low latency and horizontal scalability. This paper offered a different angle for other future research to deliberate.

ACKNOWLEDGMENT

The authors would like to thank Advanced Informatics School (AIS), Universiti Teknologi Malaysia (UTM) for the support of the resources. This work is currently funded by GUP Tier 2 from UTM (Vot Number:14H08)

REFERENCES

[1] R. Kimball and M. Ross, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. John Wiley & Sons, 2011.

[2] B. Ellis, Real-Time Analytics: Techniques to Analyze and Visualize Streaming Data. John Wiley & Sons, 2014.

[3] S. K. Bansal and S. Kagemann, "Integrating Big Data: A Semantic Extract-Transform-Load Framework," Computer, vol. 48, no. 3, pp. 42–50, Mar. 2015.

[4] R. Kimball and M. Ross, The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence. Wiley Publishing, 2010.

[5] A. Wibowo, "Problems and available solutions on the stage of Extract, Transform, and Loading in near real-time data warehousing (a literature study)," in 2015 International Seminar on Intelligent Technology and Its Applications (ISITIA), 2015, pp. 345–350.

[6] T. Jain, R. S, and S. Saluja, "Refreshing Datawarehouse in Near Real-Time," Int. J. Comput. Appl., vol. 46, no. 18, pp. 24–29, May 2012.

[7] J. Zuters, "Near Real-Time Data Warehousing with Multi-stage Trickle and Flip," in Perspectives in Business Informatics Research, 2011, pp. 73–82.

[8] Z. Lin, Y. Lai, C. Lin, Y. Xie, and Q. Zou, "Maintaining Internal Consistency of Report for Real-Time OLAP with Layer-Based View," in Web Technologies and Applications, 2011, pp. 143–154.

[9] "ETL Evolution for Real-Time Data Warehousing," TechRepublic. [Online]. Available: http://www.techrepublic.com/resource-library/whitepapers/etl-evolution-for-real-time-data-warehousing/. [Accessed: 30-Mar-2017].

[10] R. S, S. B. B, and N. K. Karthikeyan, "From Data Warehouses to Streaming Warehouses: A Survey on the Challenges for Real-Time Data Warehousing and Available Solutions," Int. J. Comput. Appl., vol. 81, no. 2, pp. 15–18, Nov. 2013.

[11] F. Majeed, M. S. Mahmood, and M. Iqbal, "Efficient data streams processing in the real time data warehouse," in 2010 3rd International Conference on Computer Science and Information Technology, 2010, vol. 5, pp. 57–61.fdsfsd

[12] R. P. Deb Nath, K. Hose, T. B. Pedersen, and O. Romero, "SETL: A programmable semantic extract-transform-load framework for semantic data warehouses," Inf. Syst.

[13] D. Ostrowski, N. Rychtyckyj, P. MacNeille, and M. Kim, "Integration of Big Data Using Semantic Web Technologies," in 2016 IEEE Tenth International Conference on Semantic Computing (ICSC), 2016, pp. 382–385.

[14] S. Fang et al., "An Integrated System for Regional Environmental Monitoring and Management Based on Internet of Things," IEEE Trans. Ind. Inform., vol. 10, no. 2, pp. 1596–1605, May 2014.

[15] X. Li and Y. Mao, "Real-Time data ETL framework for big real-time data analysis," in 2015 IEEE International Conference on Information and Automation, 2015, pp. 1289–1294.

[16] “Addressing BI Transactional Flows in the Real-Time Enterprise Using GoldenGate TDM - (Industrial Paper) - Semantic Scholar.” [Online]. Available: /paper/Addressing-BI-Transactional-Flows-in-the-Real-Time-Pareek/10c635702e00476d81a6a4d3fb29c336659d7f10. [Accessed: 31-Mar-2017].

[17] N. Narendra, K. Ponnalagu, S. Tamilselvam, and A. Ghose, “Goal-driven context-aware data filtering in IoT-based systems,” Fac. Eng. Inf. Sci. - Pap. Part A, pp. 2172–2179, Jan. 2015.

[18] C. R. Valencio, M. H. Marioto, G. F. D. Zafalon, J. M. Machado, and J. C. Momente, “Real Time Delta Extraction Based on Triggers to Support Data Warehousing,” DeepDyve, Dec. 2013.\

[19] F. Dehne, Q. Kong, A. Rau-Chaplin, H. Zaboli, and R. Zhou, “A distributed tree data structure for real-time OLAP on cloud architectures,” in 2013 IEEE International Conference on Big Data, 2013, pp. 499–505